An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 37 (1999) 171–176

Drawing a Tree on Parallel Lines

R. HIRABAYASHI AND Y. T. IKEBE
Department of Management Sciences, Science University of Tokyo
Kagurazaka, Shinjuku 162, Japan

K. IWAMURA AND T. NAKAYAMA
Department of Mathematics, Josai University
Sakado, Saitama 350-02, Japan

Abstract—We consider a problem of drawing a tree on parallel lines. In this problem we are given a tree and an infinite number of parallel lines in the plane. The object is to draw the tree so that

(i) each vertex is placed on one of the given parallel lines,
(ii) no two edges intersect, and
(iii) the 'height' of each vertex is nondecreasing, while minimizing the total number of lines used.

We show that this problem is solvable in time linear on the size of the tree, by presenting an algorithm which solves it recursively.

1. INTRODUCTION

Suppose we are given a tree on n vertices, and an (infinite) set of parallel lines in the plane. We wish to draw the tree in the plane so that

(i) each vertex is placed on one of the parallel lines,
(ii) the line segments corresponding to edges of the tree do not intersect each other, and
(iii) the 'height' of each vertex is nondecreasing along any path beginning from a vertex placed the 'lowest' line.

The object is to use the fewest number of lines possible.

This problem was first posed in [1] by Fukuhara. In this paper, we present a simple recursive algorithm to solve this problem. Moreover, the algorithm runs in time linear on the number of vertices of the tree.

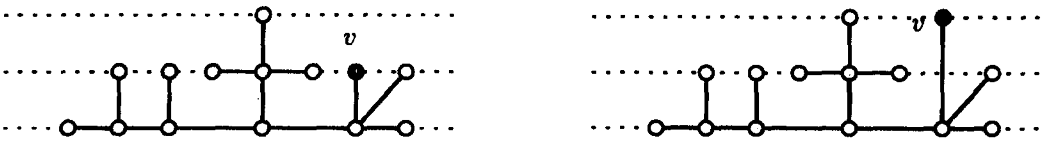
2. THE ALGORITHM

Let T be a tree on n vertices. For any vertex v of T, the *degree* of v is the number of edges incident to v, and is denoted by $\delta_T(v)$. Vertices with degree one are called *leaves*, all other vertices are called *internal*. A tree with only one internal vertex is called a *star*.

Given T and a set of parallel lines $\{\ell_1, \ell_2, \dots\}$, we wish to draw T in the plane so that

(i) each vertex is placed on one of the parallel lines; we will call the number of the line which vertex v is placed upon the *level* of v,
(ii) the line segments corresponding to edges of T do not intersect each other (except possibly at their endpoints), and
(iii) the level of vertices is nondecreasing along any path beginning from a vertex having the lowest level.

0898-1221/99/$ - see front matter. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(99)00155-8



(a) Standard optimal drawing. (b) Nonstandard optimal drawing.

Figure 1. Standard and nonstandard optimal drawings.

We will call such a drawing *feasible*. The problem is to find a feasible drawing of T in which the number of lines on which vertices are placed is minimum. We will call such a feasible drawing *optimal*. Of course, we may assume without loss of generality that there is always a vertex on line ℓ_1 in any feasible drawing. Hence, we may restate an optimal drawing as a feasible drawing in which the maximum level over all vertices is minimized. Obviously, there may be many optimal drawings of the same tree. For example, Figures 1a and 1b both depict optimal drawings of the same tree. However, the drawing in Figure 1a is obviously more 'natural' than the drawing in Figure 1b, and the reason is that in Figure 1b, the level of vertex v is unnecessarily high. In order to eliminate such 'unnatural' drawings, we introduce the concept of standard optimal drawings. More explicitly, we say that an optimal drawing is *standard* if the level of any vertex is minimum, when the levels of all other vertices are fixed. In the sequel, we observe some properties of standard optimal drawings, and use these properties to construct a standard optimal drawing for any given tree. Before going into details, we first note that

(a) any vertex with degree two which is either adjacent to a leaf, or adjacent to another vertex of degree two may be contracted, and
(b) if T is a star or path, then T may be trivially drawn using two, respectively, one lines.

Here by contracting a vertex v with degree two, we mean we replace the vertex v and the two edges (u, v) and (v, w) incident to v, by the edge (u, w). Note that this method of contraction ensures that if two internal vertices with degree at least three are adjacent to each other after performing such contractions, then they must have been adjacent to each other in the original tree. Henceforth, we will assume that any internal vertex of T adjacent to a leaf has degree at least three, there are no consecutive vertices of degree two, and that T contains a path of length three or more. We also state the following, rather obvious facts.

FACT 1. The vertices of level one form a path.

FACT 2. In a standard optimal drawing of T, there are at least two vertices of level one.

We now subdivide the internal vertices of T. For any internal vertex v of T, we say that v is a *semileaf* if the number of leaves adjacent to v is equal to $\delta_T(v) - 1$. The following lemmas pertaining to semileaves hold.

LEMMA 2.1. *Let v be a semileaf of T with $\delta_T(v) \geq 4$. Then, in any standard optimal drawing of T, the levels of all leaves adjacent to v are either equal to, or one more than that of v, and there exists a leaf whose level is exactly one more than that of v.*

PROOF. From Fact 2, the level of a leaf adjacent to v can never be less than that of v, in any feasible drawing. However, at most only two vertices adjacent to v can be placed on the same level as v, hence, there always exists at least one leaf whose level is more than the that of v. On the other hand, it is easy to see that all leaves adjacent to v can be placed on the next level, and the statement follows (see Figure 2). ∎

LEMMA 2.2. *Let v be a semileaf of T with $\delta_T(v) = 3$. Then, in any standard optimal drawing of T, the levels of the two leaves adjacent to v are either both equal to the level of v, or the level of one leaf is equal to the level of v, and the level of the other is equal to the level of v plus one.*

PROOF. As in the previous lemma, the level of a leaf adjacent to v must always be at least that of v, in any feasible drawing. Now let u be the unique internal vertex adjacent to v, and consider

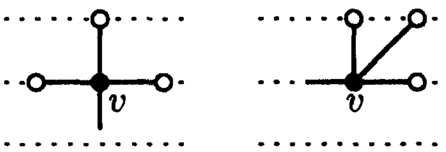


Figure 2. Drawing h2-leaves.

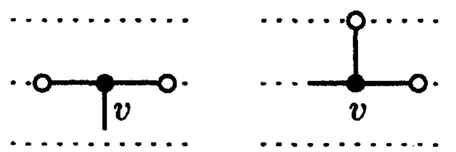


Figure 3. Drawing quasi-h2-leaves.

any optimal drawing. If the level of u is not the same as that of v, then both leaves adjacent to v can be placed on the same level. If the level of u is equal to the level of v, then only one of the leaves can be on the same level, and the other must be placed on the next level (see Figure 3). ∎

In light of these lemmas, we will call a semileaf with degree at least four, an *h2-leaf* (for height-two leaf), and a semileaf with degree three, a *quasi-h2-leaf*. These lemmas say that we may find an optimal drawing of T from the tree T' obtained by shrinking all leaves adjacent to a semileaf of T into that semileaf. Here, by shrinking a leaf into the internal vertex v to which it is adjacent, we mean that we delete the leaf and its unique incident edge, and make v 'remember' this deletion (perhaps by maintaining a list containing this leaf). Figure 6b shows the tree T' resulting from these shrinkings of the tree T shown in (a) of the same figure. For any leaf v of this new tree T', exactly one of the following cases must occur.

(a) v is also a leaf of T, and the unique vertex adjacent to v (in both T and T') is not a semileaf in T.
(b) v is an h2-leaf in T.
(c) v is a quasi-h2-leaf in T.

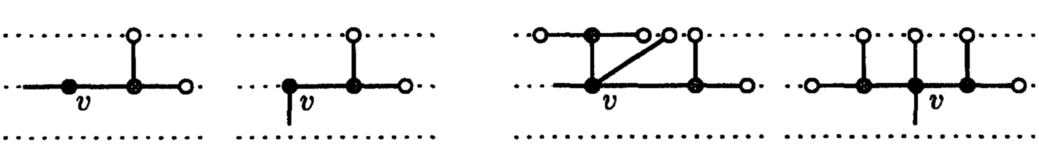
We now observe some properties of semileaves in T'. Note that for any semileaf v of T', at least one of the leaves adjacent to v must be an h2-leaf or a quasi-h2-leaf.

LEMMA 2.3. *Let v be a semileaf of T' for which the adjacent leaves contain no h2-leaf of T, and let U be the set of vertices consisting of the leaves of v, and any leaves of T which have been shrunk into them. Then, for any standard optimal drawing of T, the level of any vertex of U is either equal to the level of v, or the level of v plus one, and there is at least one vertex which attains the latter value.*

PROOF. (See Figures 4 and 5.) We distinguish the two cases: $\delta_{T'}(v) = 2$ (Figure 5a) and $\delta_{T'}(v) \geq 3$ (Figure 5b). In each case, it is easily seen that the statement holds. ∎

(a) Semi–leaf v of T'. (b) The set U.

Figure 4. The set U consisting of leaves of v and vertices shrunken into them.



(a) $\delta(v) = 2$. (b) $\delta(v) \geq 3$.

Figure 5. Drawing a semileaf v of T' with no adjacent h2-leaves.

Lemma 2.3 says that any semileaf of T' which has no h2-leaves adjacent to it may be shrunk along with its adjacent leaves into an h2-leaf. Let T'' be the tree obtained from T' by performing

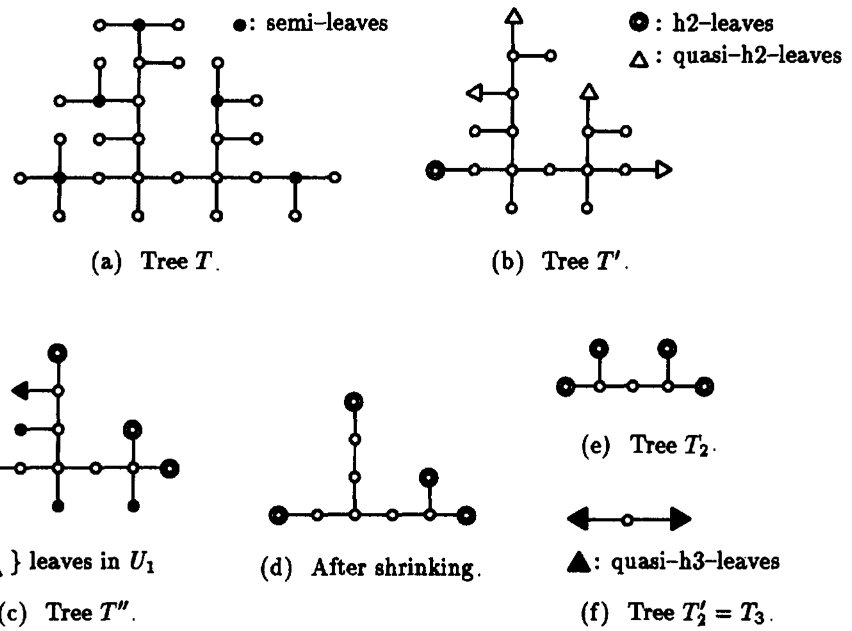


Figure 6. Creating T_2 from T.

all such shrinkings. Figure 6c shows the T'' of the previous example. It is easily seen that any semileaf of T'' always has an h2-leaf adjacent to it.

LEMMA 2.4. *Let U_1 be the set of leaves of T'' which are not h2-leaves, together with any vertices which have been shrunk into them; and likewise, let U_2 be the set of h2-leaves, and any shrunken vertices. Then, the level of any vertex of U_1 is less than or equal to the maximum level of the vertices in U_2 (see Figure 6c).*

Roughly speaking, this lemma says that we may shrink all leaves of T'' which are not h2-leaves into the internal vertices to which they are adjacent. If we do this, this will result in a tree whose leaves are all h2-leaves, and which may have many internal vertices of degree two (see Figure 6d). Moreover, an optimal drawing of the original tree T can easily be constructed from an optimal drawing of this resulting tree, hence, it suffices to find such an optimal drawing. To do this, we contract such vertices of degree two as allowed, to obtain the tree T_2 (Figure 6e), and simply perform the same operations recursively to obtain T_3 (Figure 6f), except that, instead of h2-leaves and quasi-h2-leaves, we have h3-leaves and quasi-h3-leaves. And so in general, we proceed, creating $T_4, T_5, \dots$, calling semileaves of T_k either h$(k+1)$-leaves or quasi-h$(k+1)$-leaves, until we finally arrive at a path or star. This procedure may be formulated as follows.

Step 0. Set $T_1 := T$ and $k = 1$.

Step 1. If T_k is a path or star, then stop.

Step 2. Shrink all leaves adjacent to semileaves of T_k; a semileaf v with $\delta_{T_k}(v) \geq 4$ is an h$(k+1)$-leaf, a semileaf v with $\delta_{T_k}(v) = 3$ is a quasi-h$(k+1)$-leaf. Call this tree T_k'.

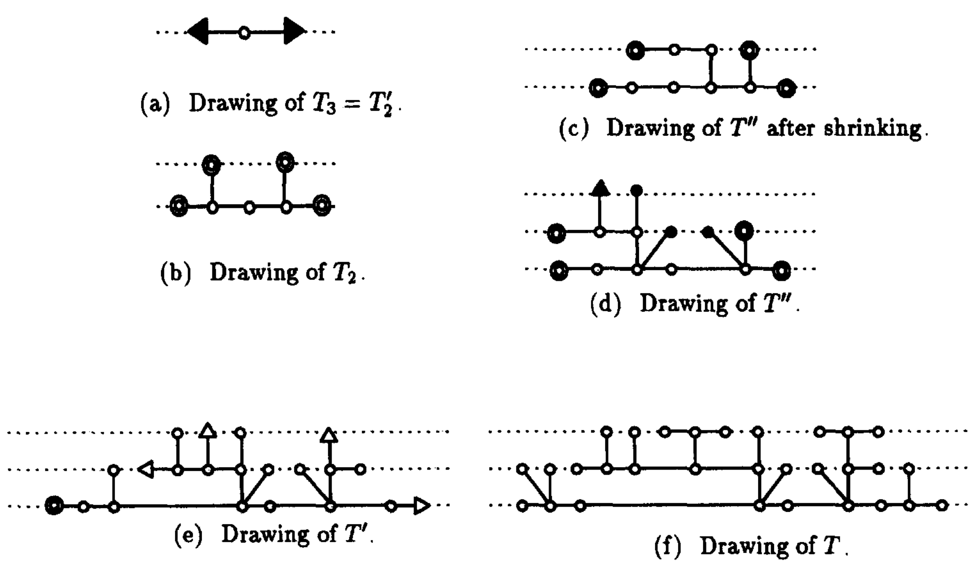
Step 3. If a semileaf of T_k' has no adjacent h$(k+1)$-leaf, shrink all its adjacent leaves, and make it an h$(k+1)$-leaf. This tree is called T_k''.

Step 4. Shrink all non-h$(k+1)$-leaves of T_k'' and contract such vertices of degree two as allowed to obtain T_{k+1}. $k := k+1$ and goto Step 1.

To obtain a standard optimal drawing from the final T_k (which is either a path or a star), we draw T_k in the obvious way, and then retrace the above procedure in reverse, expanding and drawing in the appropriate manner (see Figure 7).

From the previous discussions, the following is easily seen.

LEMMA 2.5. *For any $k > 1$, all leaves of T_k are hk-leaves (except perhaps, for the final T_k). Moreover, if v is an hk-leaf of T_k, then in any standard optimal drawing, the level of any vertex*



(a) Drawing of $T_3 = T_2'$.

(b) Drawing of T_2.

(c) Drawing of T'' after shrinking.

(d) Drawing of T''.

(e) Drawing of T'.

(f) Drawing of T.

Figure 7. Optimally drawing T.

shrunken into v is less than or equal to the level of v plus $k-1$, and there is always a vertex which attains this maximum.

Hence, the described method correctly finds an optimal drawing. We now evaluate the time complexity of this method.

LEMMA 2.6. *T_{k+1} can be obtained from T_k in $O(|T_k|)$ time, where $|T_k|$ is the number of vertices of T_k. Also, a standard optimal drawing of T_k can be obtained from one of T_{k+1} in $O(|T_k|)$ time.*

PROOF. We evaluate Steps 2, 3, and 4 separately. Step 2 can be accomplished by one execution of depth-first search [2] from an arbitrary vertex: when the search reaches a leaf, it puts a marker on its parent vertex. And when the search leaves an internal vertex, it counts the number of markers on it, compares it with its degree (which can also be computed simultaneously), and if necessary, shrinks all adjacent leaves accordingly. The shrunken vertices can be efficiently recorded by creating pointers [3] from the semileaf. Steps 3 and 4 are similar; in fact, they can both be executed during the same depth-first search used in Step 2. Since depth-first search takes $O(|T_k|)$ time, the total time spent on Steps 2, 3, and 4 is $O(|T_k|)$. Now, it can be seen that the expanding and drawing can also be done in $O(|T_k|)$ time by properly maintaining pointers to shrunken and contracted vertices. ∎

Next, we bound the size of T_k.

LEMMA 2.7. $|T_{k+1}| \leq (2/3)|T_k|$.

PROOF. We first note that $|T_{k+1}| \leq |T_k| -$ # of leaves of T_k, and that for each T_k, the degree of any semileaf is at least three, and there are no consecutive vertices of degree two. Hence, it suffices to show that

$$\#\text{ of leaves of } \bar{T} \geq \frac{1}{3}\,|\bar{T}|$$

for all trees $\bar{T}$ satisfying the above conditions. We prove this by induction on the number of vertices of $\bar{T}$. If $|\bar{T}| \leq 3$, or if $\bar{T}$ has only one internal vertex, i.e., $\bar{T}$ is a star, then the statement is obvious. Now suppose $|\bar{T}| > 3$, and $\bar{T}$ has at least two internal vertices. Let v be an arbitrary semileaf of $\bar{T}$, and let u be the unique internal vertex adjacent to v. If u has degree at least three, then let $\tilde{T}$ be the tree obtained by deleting all leaves adjacent to v, otherwise (u has degree two) let $\tilde{T}$ be the tree in which v and all leaves of v have been deleted. Then, $\tilde{T}$ itself is a tree on either $|\bar{T}| - l$ or $|\bar{T}| - (l+1)$ vertices, where l is the number of leaves adjacent to v. Since $\tilde{T}$ clearly satisfies the required conditions, we may apply the induction hypothesis to conclude that the number of leaves of $\tilde{T}$ is at least $(1/3)|\tilde{T}|$, and all of them except one (either v or u) are also leaves of $\bar{T}$. Hence,

$$\# \text{ of leaves of } \bar{T} = \# \text{ of leaves of } \tilde{T} - 1 + l \geq \frac{1}{3}\left(|\bar{T}| - l - 1\right) - 1 + l = \frac{1}{3}|\bar{T}| + \frac{2}{3}l - \frac{4}{3} \geq \frac{1}{3}|\bar{T}|,$$

because $l \geq 2$. ∎

Combining these, we obtain the following.

THEOREM 2.8. *The proposed algorithm correctly finds a standard optimal drawing in $O(n)$ time.*

PROOF. The correctness of the algorithm has already been shown, so it suffices to establish the time complexity. From Lemma 2.6, the total running time is bounded by

$$O(|T|) + O(|T_2|) + \cdots,$$

and $|T_k| \leq (2/3)^k |T|$ from Lemma 2.7, so the total complexity is of $O(|T|) = O(n)$. ∎

REFERENCES

1. T. Fukuhara, Structural study of a graph drawing problem on k parallel lines, Master Thesis (in Japanese), Dept. of Engineering and Electronics, Tokyo Institute of Technology, Tokyo, (1990).
2. R. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* **1**, 146–160 (1972).
3. T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms*, MIT Press, Boston, MA, (1990).